

AUTOMATIC SPEECH RECOGNITION USING HIDDEN CONDITIONAL NEURAL FIELDS

Yasuhisa Fujii, Kazumasa Yamamoto, Seiichi Nakagawa

Toyohashi University of Technology
Department of Computer Science and Engineering
1-1 Hibarigaoka, Tenpaku-cho, Toyohashi, Aichi 441-8580, JAPAN

ABSTRACT

Hidden Conditional Random Fields(HCRF) is a very promising approach to model speech. However, because HCRF computes the score of a hypothesis by summing up linearly weighted features, it cannot consider non-linearity among features that will be crucial for speech recognition. In this paper, we extend HCRF by incorporating gate function used in neural networks and propose a new model called Hidden Conditional Neural Fields(HCNF). Differently with conventional approaches, HCNF can be trained without any initial model and incorporate any kinds of features. Experimental results of continuous phoneme recognition on TIMIT core test set and Japanese read speech recognition task using monophone showed that HCNF was superior to HCRF and HMM trained in MPE manner.

Index Terms— hidden conditional neural fields, hidden conditional random fields, HMM, speech recognition

1. INTRODUCTION

Current ASR systems employ Hidden Markov Model(HMM) with Gaussian Mixture Model(GMM) as emission probability for acoustic model. However, HMM has two major drawbacks to use for acoustic model. Firstly, HMM has a strong independency assumption that frames are independent given a state, thus, lacks an ability to deal with a feature which straddles over several frames. Secondly, because HMM is a generative model, it is not suitable to discriminate sequences. To solve the former problem, features that can deal with phenomena straddling over frames have been developed such as delta coefficient [1], segmental statistics [2] and modulation spectrum [3]. For the latter problem, discriminative training methods like MPE have been investigated [4].

Combining a model that can consider long range features and has high discriminative power with HMM has also been conducted to tackle above problems. For example, Tandem system extracts features using Multi Layered Perceptron(MLP) and utilizes them as an input for HMM [5]. Conditional Random Fields(CRF) [6] also could be used instead of MLP [7].

Hidden Conditional Random Fields (HCRF) is a very promising approach to model speech [8, 9, 10]. HCRF overcomes two major drawbacks of HMM mentioned above while

preserving the merits of HMM such as efficient algorithms including forward-backward algorithm and Viterbi decoding. Although HCRF is promising, it still has a drawback that it cannot consider non-linearity among features which would be crucial for speech recognition because it computes the score of a hypothesis by summing up linearly weighted feature values. There is an attempt to expand feature vectors explicitly [11]. However, it will be difficult to use the approach when the number of features increases.

Peng et al. proposed Conditional Neural Fields(CNF) which can consider non-linearity among features by incorporating gate function into CRF[12]. In this paper, we propose Hidden Conditional Neural Fields(HCNF) which can consider non-linearity between features by introducing gate function into HCRF. Differently with conventional approaches, HCNF can be trained without any initial model and incorporate any kinds of features. Although it seems that our work is close to the work presented in [13], their work extended CRF to incorporate gate function, and thus, it was close to CNF rather than our work. Experimental results on continuous phoneme recognition on TIMIT core test set and Japanese read speech recognition task show that HCNF is superior to HCRF and HMM trained in MPE manner.

This paper is organized as follows. In the next section, HCRF for ASR is described. After that, HCNF is introduced in Section 3. In Section 4, the training methods for HCRF and HCNF are explained. The experimental setup and results are shown in Sections 5. Finally, Section 6 presents our conclusions and some future works.

2. HIDDEN CONDITIONAL RANDOM FIELDS

2.1. Formulation

Given a observation sequence $X = (x_1, x_2, \dots, x_T)$, HCRF computes a score of a label sequence $Y = (y_1, y_2, \dots, y_T)$ as follows:

$$P(Y|X) = \frac{1}{Z(X)} \sum_S \exp(\kappa(\Phi_r(X, Y, S) + \Psi_r(X, Y, S))), \quad (1)$$

where κ is a state-flattening coefficient [14]. $Z(X)$ is a partition function and computed as follows:

$$Z(X) = \sum_Y \sum_S \exp(\kappa(\Phi_r(X, Y, S) + \Psi_r(X, Y, S))), \quad (2)$$

This work was supported by Global COE Program Frontiers of Intelligent Sensing from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

$\Phi_r(X, Y, S)$ and $\Psi_r(X, Y, S)$ are an obseration function and a transition function, respectively, and defined as follows:

$$\Phi_r(X, Y, S) = \sum_k w_k \sum_t \phi_k(X, Y, S, t), \quad (3)$$

$$\Psi_r(X, Y, S) = \sum_j u_j \sum_t \psi_j(X, Y, S, t, t-1). \quad (4)$$

$S = (s_1, s_2, \dots, s_T)$ is a hidden variable sequence that represents a state sequence. $\phi_k(X, Y, S, t)$ and $\psi_j(X, Y, S, t, t-1)$ mean a raw observation feature extracted at frame t and a transition feature extracted at frame t and $t-1$, respectively. Corresponding weights are w_k and u_j , respectively. Because the feature functions used in HCRF are restricted to current frame in the observation function, and current and previous frames in the transition function, efficient algorithms such as forward-backward algorithm and Viterbi algorithm can be adapted to HCRF.

2.2. Training

Given training data $D = \{X^i, Y^i\}, i = 0, \dots, N$, the objective function of HCRF is set as follows:

$$l(\lambda; D) = - \sum_i \log P(Y^i | X^i). \quad (5)$$

$\lambda = \{w_k, u_j\}$ which minimizes $l(\lambda; D)$ can be found using gradient based methods such as L-BFGS and Stochastic Gradient Descent(SGD). The partial derivatives of Eq.(5) with respect to w_k can be computed as follows:

$$\frac{\partial l(\lambda; D)}{\partial w_k} = - \kappa \sum_i E \left[\sum_t \phi_k(X^i, Y^i, S, t) \right]_{S|Y^i, X^i} + \kappa \sum_i E \left[\sum_t \phi_k(X^i, Y, S, t) \right]_{Y, S|X^i}, \quad (6)$$

where $E[\cdot]_X$ means expectation by X . The partial derivatives with respect to u_j can be derived as well as Eq.(6).

2.3. Inference

[10] used N-best inference to find the sequence that maximizes Eq.(1). Although the method yileded a reasonable result, it will be difficult to adapt it to LVCSR directly. Therefore, we decided to use Viterbi algorithm for inference. This means that hidden variable S is not marginalized out in inference.

3. HIDDEN CONDITIONAL NEURAL FIELDS

3.1. Formulation

HCRF is an extention of HCRF by introducing gate function into it. Fig.1 shows the structures of HCRF and HCNF. HCNF uses a different observation function while using the same transition function with HCRF as follows:

$$\Phi_n(X, Y, S) = \sum_t \sum_g^K w_{y_t, s_t, g} h(\theta_{y_t, s_t, g}^T \phi(X, Y, S, t)) \quad (7)$$

$$\Psi_n(X, Y, S) = \sum_j u_j \sum_t \psi_j(X, Y, S, t, t-1) \quad (8)$$

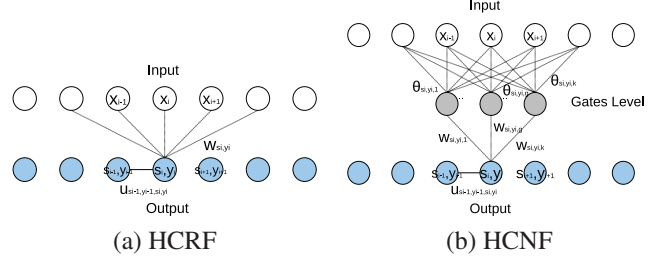


Fig. 1. The structure of HCRF and HCNF.

where $\phi(X, Y, S, t)$ is a vector representation of $\phi_k(X, Y, S, t)$ in Eq.(3) and $\theta_{y,s,g}$ is a corresponding weight vector specific to the triple of y, s and g ¹, $h(x)$ is a gate function defined as follows:

$$h(x) = \frac{1}{1 + \exp(-x)} - 0.5. \quad (9)$$

In HCNF, the observation function $\Phi_n(X, Y, S)$ uses K gate functions by which it considers non-linearity among features. Using Eqs.(7) and (8), $P(Y|X)$ can be computed as follows:

$$P(Y|X) = \frac{1}{Z(X)} \sum_S \exp(\kappa(\Phi_n(X, Y, S) + \Psi_n(X, Y, S))), \quad (10)$$

where $Z(X)$ is a partition function and computed as follows:

$$Z(X) = \sum_{Y'} \sum_S \exp(\kappa(\Phi_n(X, Y', S) + \Psi_n(X, Y', S))). \quad (11)$$

Because the feature functions used in HCNF are also restricted to a current frame in the observation function, and current and previous frames in the transition function, as well as HCRF, efficient algorithms such as forward-backward algorithm and Viterbi algorithm can be also adapted to HCNF.

3.2. Training

Similar to HCRF, the objective function of HCNF is defined as follows:

$$l(\lambda; D) = - \sum_i \log P(Y^i | X^i). \quad (12)$$

The partial derivatives of Eq.(12) with respect to $w_{y,s,g}$ and $\theta_{y,s,g}$ can be computed as follows:

$$\frac{\partial l(\lambda; D)}{\partial w_{y,s,g}} = - \kappa \sum_i E \left[\sum_t h(\theta_{y,s,g}^T \phi(X^i, Y^i, S, t)) \right]_{S|Y^i, Y^i} + \kappa \sum_i E \left[\sum_t h(\theta_{y,s,g}^T \phi(X^i, Y, S, t)) \right]_{Y, S|X^i}, \quad (13)$$

¹By this definition, gate functions are dependent on their states unlike original CNF [12]. We employed this definition because it was robust in our initial experiments. However we can obtain state independent gate functions by setting $\theta_{y_t, s_t, g} \triangleq \theta_g$.

$$\frac{\partial l(\lambda; D)}{\partial \theta_{y,s,g}} = -\kappa \sum_i E \left[\sum_t w_{y,s,g} \frac{\partial h(\theta_{y,s,g}^T \phi(X^i, Y^i, S, t))}{\partial \theta_{y,s,g}} \right]_{S|X^i, Y^i} + \kappa \sum_i E \left[\sum_t w_{y,s,g} \frac{\partial h(\theta_{y,s,g}^T \phi(X^i, Y, S, t))}{\partial \theta_{y,s,g}} \right]_{Y, S|X^i}. \quad (14)$$

The devivative of Eq.(9) can be computed as follows:

$$\frac{dh(x)}{dx} = (0.5 + h(x))(0.5 - h(x)). \quad (15)$$

The partial derivatives with respect to u_j can be derived as well as Eq.(6).

3.3. Inference

Just like in the case of HCRF, Viterbi algorithm can be used to find the most likely sequence of hidden states instead of searching for the most likely output sequence that maximizes (10).

4. TRAINING METHOD

4.1. Regularization

Regularization is effective to avoid overfitting in HCRF training [9]. Therefore, we regularize the objective function of both HCRF and HCNF as follows:

$$f(\lambda; D) = l(\lambda; D) + r(\lambda) \quad (16)$$

where $r(\lambda)$ is L1 or L2 regularization:

$$\text{L1: } r(\lambda) = C \|\lambda\|_1 = C \sum_i |\lambda_i| \quad (17)$$

$$\text{L2: } r(\lambda) = C \|\lambda\|_2 = \frac{C}{2} \sum_i \lambda_i^2 \quad (18)$$

where C is a trade-off between $l(\lambda; D)$ and the regularization term.

4.2. Optimization Algorithm

We adopted Stochastic Gradient Descent (SGD) to train HCRF and HCNF. SGD is a kind of online algorithm and has favorable behaviors that it does not likely fall into local optima and it has quite fast convergence speed. SGD updates parameters using a gradient g_t computed from subset of entire training data in each iteration:

$$\lambda_{t+1} = \lambda_t - \eta_t g_t, \quad (19)$$

where η_t is a learning rate. In this paper, we compute g_t from only one sample (utterance). Alternative to choosing a sample randomly from training data, we shuffled entire training data and use the samples for training in the same order in all iterations over the data. Suppose $\#iter$ means the number of times entire training data was used and $\#sample$ means the

number of samples in training data, then, η_t is computed as follows:

$$\eta_t = \alpha \frac{\#sample \cdot \#iter - t}{\#sample \cdot \#iter}. \quad (20)$$

where α determines the value of η_0 . We used FOBOS for SGD with regularization [15]

5. EXPERIMENTS

5.1. Setup

We used the TIMIT corpus to examine the effectiveness of our proposed method because it offers a good test bed to study algorithmic improvements [16]. The training set in the TIMIT corpus consists of 3696 utterances by 462 speakers ($\approx 3h$). For evaluation, we used the core test set consisting of 192 utterances by 24 speakers. We also used the ASJ+JNAS corpus² which is about 11 times larger than the TIMIT corpus. The training set in the ASJ+JNAS corpus consists of 20337 by 133 speakers ($\approx 33h$). For evaluation we used the IPA100 test set consisting of 100 utterances by 23 speakers. We extracted 13 MFCC features and their deltas and double deltas to form 39-dimensional observation for the TIMIT corpus. Log energy was used instead of 0th MFCC for the ASJ+JNAS corpus. The speech was analyzed using a 25ms Hamming window with pre-emphasis coefficient 0.97 and shifted with a 10ms fixed frame advance. For the TIMIT corpus, the 61 TIMIT phonemes were mapped into 48 phonemes for training and further collapsed from 48 phonemes to 39 phonemes for evaluation [17]. For the ASJ+JNAS corpus, 43 Japanese phonemes are used. All phonemes are represented as 3 state left-to-right monophone models. We defined three types of observation functions as follows:

$$\phi_{sb}^{M1}(X, Y, S, t) = \delta(s_t = s) x_{d,t+b}, \quad (21)$$

$$\phi_{sb}^{M2}(X, Y, S, t) = \delta(s_t = s) x_{d,t+b}^2, \quad (22)$$

$$\phi_s^{Occ}(X, Y, S, t) = \delta(s_t = s). \quad (23)$$

b is used to consider surrounding frames. In the experiments, we used $-4 \leq b \leq 4$, and therefore, we used the amount of observations of 9 frames centered at the current frame. Also, we defined a transition function as follows:

$$\psi_{ss'}^{Tr}(X, Y, S, t, t-1) = \delta(s_t = s) \delta(s_{t-1} = s'). \quad (24)$$

The M1 and M2 features were normalized to have zero mean and unit variance. All parameters of HCRF are initialized with 0 and all parameters of HCNF are initialized at random between -0.5 and 0.5. Regularization parameter C and state-flattening parameter κ were set to 1.0. The parameters of HCRF were trained by 10 iterations while the parameters of HCNF were trained by 30 iterations for the TIMIT corpus and 15 iterations for the ASJ+JNAS corpus by SGD.

For comparison, we conducted recognition experiments using HMMs which have same topology with HCNFs and HCRFs (monophone). Initially, HMM with diagonal covariance matrices and 32 mixture GMM was trained in MLE

²http://www.mibel.cs.tsukuba.ac.jp/_090624/jnas/instruct.html

6. CONCLUSION

Table 1. Phoneme recognition results on TIMIT core test set[%].

Model	Del.	Ins.	Subs.	PER
MLE-HMM (diag,32mix)	9.2	2.6	18.2	30.0
MMI-HMM (diag,32mix)	7.9	2.8	18.0	28.7
MPE-HMM (diag,32mix)	9.1	2.2	17.1	28.4
HCRF (none)	6.3	3.9	21.4	31.6
HCRF (L1)	6.6	3.6	20.6	30.8
HCRF (L2)	6.8	3.0	20.6	30.5
HCNF (none,K=4)	6.3	3.5	20.9	30.8
HCNF (L1,K=4)	7.8	2.6	18.3	28.7
HCNF (L2,K=4)	7.2	2.5	19.2	28.9
HCNF (L2,K=4, $\kappa = 0.2$)	9.2	1.5	17.3	27.9

Table 2. Phoneme recognition results on IPA100 test set[%].

Model	Del.	Ins.	Subs.	PER
MLE-HMM (diag,32mix)	6.5	2.0	11.1	19.6
MMI-HMM (diag,32mix)	5.5	1.9	8.6	16.0
MPE-HMM (diag,32mix)	6.0	1.1	8.0	15.1
HCRF (L2)	6.5	0.8	9.6	16.8
HCNF (L2,K=4)	6.5	0.7	8.0	15.2
HCNF (L2,K=4, $\kappa = 0.2$)	6.3	0.7	7.8	14.8

manner by using HTK. The MLE-HMM model was used to train MMI and MPE HMMs with I-smoothing 100, a learning rate parameter 2 and a scaling factor 0.2 (similar to the state-flattening coefficient used in HCNFs), and the parameters are updated 10 times. A bigram phone language model was trained from training corpora.

5.2. Results

Table1 shows the phoneme recognition results on the TIMIT core test set. We can see that the both regularizations consistently improved the performance of both HCRF and HCNF. The results of HCRFs were inferior to the results of HMMs and significantly worse than the result of 28.2% reported in [10]. This is because of our implementation of HCRF in which mixtures were not used unlike [10]. Instead of using mixtures in HCRF, we extended it to HCNF by introducing gate functions. HCNFs clearly outperformed HCRFs and the result showed the effectiveness of incorporating the gate function into HCRF. By setting $\kappa = 0.2$, we obtained the best performance of 27.9%. This result was superior to the results of HMMs and comparable with the best ones of previous results in monophone setting. The parameter number of HCNF was 412656 while that of HMM was 366672 (that of HCRF was 104928) and they did not have large difference³.

Table2 shows the phoneme recognition results on the IPA100 test set. We trained HCRF and HCNF models only with L2 regularization. On the test set, HCNF outperformed HCRF again. Also, HCNF outperformed HMMs when setting $\kappa = 0.2$. The results showed the scalability of HCNF since the ASJ+JNAS corpus was about 11 times larger than the TIMIT corpus.

³HMMs with a GMM of 64 mixtures did not provide any improvements over the HMMs with a GMM of 32 mixtures. Therefore, the number of parameters for HMMs was enough in this monophone setting.

In this paper, we proposed HCNF for ASR which incorporated gate functions used in neural networks into HCRF. The phoneme recognition results of HCNF on TIMIT and ASJ+JNAS corpora outperformed the results of HCRF and the results of HMMs trained in MPE manner. HCNF could be trained successfully without not only reasonable initial model but also initial alignment.

In our future, we need to examine HCNF by LVCSR experiments. A resemblant idea was examined in [18] if we use HCNF as a kind of acoustic model. However, we will adapt HCNF to LVCSR directly. Also, we want to extend it to context dependent one. Exploring the effective features for HCNF is also curious. In addition, although in this paper, we used the sigmoid function as the gate function, the performance of neural networks depends on used gate function [19]. Therefore, we will explore the impact of the gate function used in HCNF.

7. REFERENCES

- [1] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Transactions of Acoustics Speech and Signal Processing*, vol. 34, no. 1, pp. 52 – 59, Feb. 1986.
- [2] S. Nakagawa and K. Yamamoto, "Speech recognition using hidden markov models based on segmental statistics," *Systems and Computers in Japan*, vol. 28, no. 7, pp. 31–38, Jun. 1997.
- [3] N. Kanedera, T. Arai, H. Hermansky, and M. Pavel, "On the relative importance of various components of the modulation spectrum for automatic speech recognition," *Speech Communication*, vol. 28, pp. 43–55, 5 1999.
- [4] D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. thesis, Cambridge University Engineering Dept, 2003.
- [5] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature stream extraction for conventional HMM systems," in *Proc. ICASSP*, 2000.
- [6] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the 18th International Conference on Machine Learning*, 2001.
- [7] E. Fosler-L. and J. Morris, "Crandem systems: Conditional random field acoustic models for hidden markov models," in *Proc. ICASSP*, 2008, pp. 4049–4052.
- [8] A. Gunawardana, M. Mahajan, A. Acero, and J.C. Platt, "Hidden conditional random fields for phone classification," in *Proc. Interspeech*, 2005, pp. 1117 – 1120.
- [9] Y.-H. Sung, C. Boullis, C. Manning, and D. Jurafsky, "Regularization, adaptation, and non-independent features improve hidden conditional random fields for phone classification," in *Proc. ASRU*, 2007, pp. 347–352.
- [10] Y.-H. Sung and D. Jurafsky, "Hidden conditional random fields for phone recognition," in *Proc. ASRU*, 2009, pp. 107 – 112.
- [11] G. Heigold, D. Rybach, R. Schluter, and H. Ney, "Investigations on Convex Optimization Using Log-Linear HMMs for Digit String Recognition," in *Proc. ASRU*, 2009, pp. 216–219.
- [12] J. Peng, L. Bo, and J. Xu, "Conditional neural fields," in *Proc. Advances in Neural Information Processing Systems 22*, 2009, pp. 1419–1427.
- [13] R. Prabhavalkar and E. Fosler-Lussier, "Backpropagation training for multilayer conditional random field based phone recognition," in *Proc. ICASSP*, 2010.
- [14] M. Mahajan, A. Gunawardana, and Alex Acero, "Training algorithms for hidden conditional random fields," in *Proc. ICASSP*, May 2006, pp. I-273–I-276.
- [15] J. Duchi and Y. Singer, "Efficient learning using forward-backward splitting," in *Proc. NIPS*, 2009.
- [16] T. N. Sainath, B. Ramabhadran, and M. Picheny, "An exploration of large vocabulary tools for small vocabulary phonetic recognition," in *Proc. ASRU*, 2009, pp. 359 – 364.
- [17] K.-F. Lee and H.-W. HON, "Speaker-Independent Phone Recognition Using Hidden Markov Models," *IEEE Transactions of Acoustics Speech and Signal Processing*, vol. 37, no. 11, pp. 1641 – 1648, Nov. 1989.
- [18] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proc. ICASSP*, 2009, pp. 3761 – 3764.
- [19] S. M. Siniscalchi, T. Svendsen, F. Sorbello, and C.-H. Lee, "Experimental studies on continuous speech recognition using neural architectures with "adaptive" hidden activation functions," in *Proc. ICASSP*, 2010, pp. 4882–4885.